# Implementation of a WLAN DMX Server based on NDIS WLAN Miniport Driver

HeeJung Byun, Cui Feng and Sugoog Shon[1]

*Department of Information and Telecommunication Engineering,*
*The University of Suwon, San 2-2, Wau-ri,*
*Bongdam-eup, Hwasung-si, Kyunggi-do, Korea*
sshon@suwon.ac.kr

***Abstract***

*The paper presents a new server system for a wireless LAN DMX to enhance diverse LED lighting effects. In order to enhance the capability to send DMX packets to the large number of DMX client devices, a new WDMX server system is proposed based on a NDIS miniport driver. Through the NDIS miniport driver rather than raw TCP/IP socket, the throughput of the server is increased because header sections of TCP/IP stack are removed. Even, the format of Ethernet frame is modified to use a new type of addressing scheme which can address the devices by universe ID and DMX ID value. The implemented server software program which also use thread technology and a flexible data structure can support up to 10,500 DMX channels with the refresh rate of 40msec. Experiment shows the results and performances about the WDMX server system. Last but not least, the WDMX server system shows possibilities which can be supported as many as we want to play in entertainment area.*

***Keywords:*** *WLAN, NDIS miniport, LED, Lighting effects, WDMX server*

## 1. Introduction

Of the latest lighting technologies, LED lighting devices are used frequently in various events and entertainments. LED lighting is becoming more common from street and roadway illumination to commercial signage. Installation of lighting facilities is increased in various buildings at home and abroad to realize exterior lighting at night, secure status as local landmarks, and promote buildings. Use of LED lighting is also expanding in entertainment. LED lighting has played prominent role in the London 2012 Olympic Games, especially in the opening ceremony. 70,500 'paddles' each containing nine high powered LED lights were used to turn the huge audience into a gigantic screen. Each contained lights that could be programmed to create patterns, text and animated effects in the crowd sections – allowing organizers to create amazing video images as if each were a pixel in a TV screen [1].

Recently, LED lighting control technology has been developed to produce diversified lighting effects. For more various effects, more LED devices are connected and controlled via communication systems. Of LED remote-control methods, DMX512 communication protocol is frequently used as a standard in the lighting device-based directing effect creating sectors such as theaters, stages, and performances. It departs from control switches of lighting devices to create diversified directing effects through the use of various functions including dimmer, program control, pan, tilt, shutter, and

[1] Corresponding author

timer. In the London 2012 Olympic Games, DMX512 protocol was also used to control LED devices via communication.

Advanced DMX lighting devices need to have more DMX channels. Existing lighting devices required only about four channels in DMX frame for pan, tilt, color, and shade. Recently, diversified functions including shutter, directional change, prism, frosting, speed, dual color, lamp striking, operation mode selection, and reset have been added.

In order to improve lighting effects, few studies have been conducted by adopting communication technologies that allow more DMX512 devices to be connected and controlled. Also, few alternatives to DMX512 have been proposed. RS232/ RS422/ RS485, power line communication, DALI (Digital Addressable Lighting Interface), CAN (Controller Area Network), and LonWorks system have been used as communication method.

In conjunction with Ethernet, Advanced Control Network (ACN), known as next-generation control system, was developed by Entertainment Services and Technology Association (ESTA) and aims to overcome limitations in DMX512 and function [2]. It supports two-way networking and has been studied as dedicated lighting control protocol known as Advanced Dimmer Network (ADN), based on DMX512. MIDI data whose speed is 31,250 baud was converted into DMX512 to control lighting devices [3]. Transmission technology of DMX512 data in Art-Net IP-based network was developed by Artistic License based in the U.K. It converts DMX512 data into packets [4]. In addition, portable wireless LAN-based WDMX that can be expandable to overcome limitations in DMX512 controlled in the wired environment was developed [5]. The first commercial wireless DMX512 system was based on frequency-hopping spread spectrum (FHSS) technology using commercial wireless modems [6]. Some later WiFi/WLAN technology is used in DMX. Other studies were conducted to develop ZDMX that is expanded based on Zigbee [7]. Remote Device Management (RDM), ratified by ESTA in 2006, is a protocol enhancement to DMX512 that allows two-way communication between a lighting system controller and attached RDM compliant devices over a standard DMX line [8]. In 2011, ESTA has merged with PLASA, a similar organization in the UK. The new organization is called PLASA. Now, PLASA is the lead international membership body for those who supply technologies and services to the event, entertainment and installation industries [9].

However, some perceived limitations such as the maximum slot count of 512 per universe (each DMX network) have been addressed for the DMX standard. A universe only contains up to 512 channels and a single DMX cable is allowed for one universe. The limited number of DMX channels becomes a stumbling block to improve lighting effects. To get expansion of DMX network, special system such as DMX server or controllers are required.

This paper presents a new WiFi-based WDMX server system that can expand DMX512 channels as many as wanted and that is compatible with DMX512. It is intended to achieve spatial expansion and mobility based on wireless 802.11 LAN. This server can send WDMX packets to WDMX controllers and corresponding DMX devices for each universe. NDIS (Network Driver Interface Specification) WLAN miniport driver is used to transmit DMX data through Network interface card (NIC). Chapter 2 describes the necessity of improvement of lighting effects, limitations in DMX512 standard and proposes a WDMX server system and its packet architecture. Chapter 3 describes about implementation of a WDMX server system, its packet structure, and software. Chapter 4 describes about experiment, results and performance. Chapter 5 elaborates on the conclusions.

## 2. Proposed WDMX512 Server

The DMX protocol, referred to as Digital MultipleX, consists of a stream of data which is sent over a balanced cable system connected between the data transmitter (server) and a data receiver (client). A single DMX port, outputting this stream, can pass magnitude value information for up to 512 channels. This port is known as a DMX universe. USITT (US Institute for Theatre Technology) developed DMX protocol for dimmer control in 1986, and enhanced DMX512 protocol was officially adopted by manufacturers in 1990 [10][11]. DMX512 is a standard for digital communication networks that are used to control lighting effects.

The data stream has a specific format. DMX512 is based on RS-485 communication and uses an asynchronous 8-bit serial data with a variable-size. Each data format is fixed at one start bit, eight data bits, two stop bits and no parity. Communication frame of DMX512 has a bit stream format where 512 bytes are continued. It does not include automatic error checking and correction.

As shown in Figure 1, the start of a frame is signified by BREAK, MAB (Mark After Break). Following the break, START code (slot 0, data), channel data (512 bytes), and null are sent. START code specifies the type of data in the frame. 512 slots from 0 can be included in one packet, and several control signals are required to start a slot. It operates at a speed of 250 Kbps, and individual bit has 4μs of pulse cycle. It takes 44 μs (1 start bit + 8 data bits + 2 stop bits = 11 bits) for transmission with regard to START code and channel n.

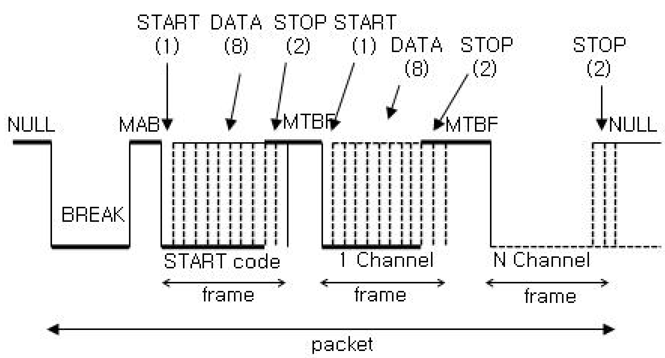


**Figure 1. A DMX512 Packet**

Individual lighting needs to be refreshed at a cycle of the minimum 44Hz for the purpose of visual lighting effects. START code informs the receivers that the packet is being refreshed. Channel 1 to channel n frames follow. Each channel is separated from the other by specified bits of start and stop data. The DMX signal is made up of a sequence of frame. After one frame is dealt with, the following frame is sent to continuously transmit a packet. If there is no change, the same data are continuously transmitted. Then it sends out a stream of serial data corresponding to the magnitude value of each channel. The mark time between frames (MTBF) can be from a little more than 0 sec to up to 1 sec. The MTBF is obviously high.

It is necessary to fulfill the refresh rate of 44Hz for 512 channels in daisy chain. For a DMX device with 3-byte channel to express colors R, G and B, the number of lighting devices is limited to 170 (≈ 512/3). If control of pan, tilt, or timer for a lighting device is taken into account, the number of lighting device is further reduced. The number of

DMX devices connected to a RS-485 bus network is traditionally limited with not more than 32 devices. If more than 32 devices need to communicate, the network can be expanded across parallel buses using DMX splitters.

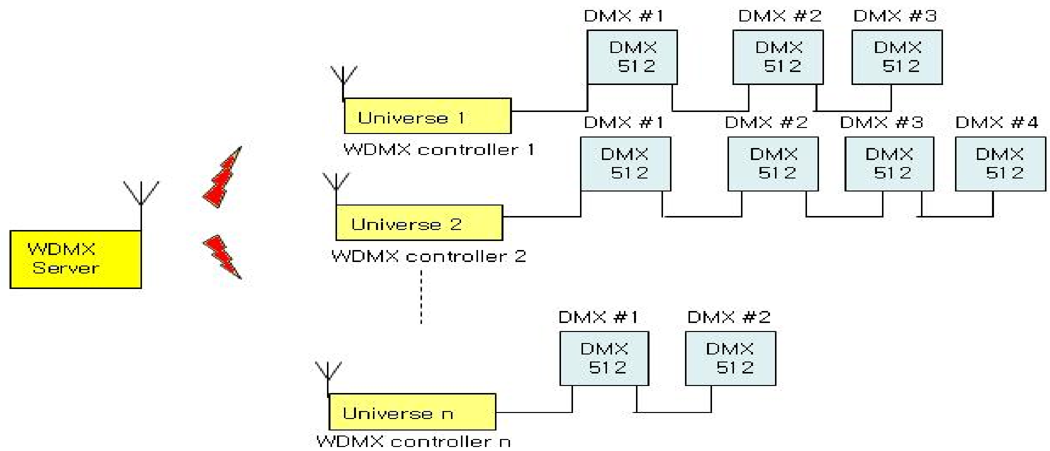


**Figure 2. The Proposed WDMX System and WDMX Server**

Figure 2 shows a WDMX lighting system that can be expanded flexibly. It consists of a WDMX server, several WDMX controllers, and DMX devices. Each of DMX devices in daisy chain is connected to a corresponding WDMX controller. The proposed WDMX server can send out DMX signal in order to produce lighting effects. Then, directing data is transmitted to a WDMX controller through wireless LAN NIC. The WDMX controller receives data from the server to convert them befitting DMX communication protocol before transmission to a lighting device. Direction to be controlled is stored in DMX controller memory of DMX device before execution. WDMX controller works like DMX splitters.

## 3. WDMX Server System

### 3.1. WDMX Server

For the proposed WDMX server, the most challenging issues become both the transmission ability enough to fulfill the requirement for the DMX refresh rate and how to expand the limitation of DMX devices per universe.

The WDMX server would communicate with several WDMX controllers. The WDMX server is the computer system with application program that can transmit WDMX packets to each WDMX controller. The server needs communication protocol that would connect with multiple WDMX controllers.

As shown in Figure 3, we design the WDMX packet format that the server would deal with. The WDMX packet is analogous to the format of Ethernet frame, but its content is not same. The WDMX packet consists of five fields such as a ‘destination MAC address’, ‘source MAC address’, ‘packet ID’, and ‘payload’, ‘CRC’, *etc*. The packet ID is the sequence number of WDMX packets that WDMX server will send out for performing lighting effects. A universe, a part of payload, is composed of four fields such as a ‘size of universe’, a ‘universe ID’, a ‘DMX data’, and a ‘check sum’. Moreover, a packet can consist of several universes. A DMX device is identified through a corresponding universe ID and DMX channel value. It is necessary to address about the universe size in order to count the number of bytes that each universe uses. Each universe holds different number of DMX channels for lighting effects. The universe ID can be assigned from 0 up to $2^{nun}-1$ with 'nun' bits. DMX data value ranges 1 to 512 bytes. The value of the universe size is determined by the sum of all the universe fields. Also, remind that the maximum slot count of DMX channels per universe should not exceed 512.

In our design, the minimum size of a universe becomes 5 bytes and the maximum becomes 516 bytes, which is equal to the sum of 'universe length (10 bits)' + 'Universe ID(nun = 14 bits)' + 'DMX Data (1-512 bytes)' + 'Check sum(8bits)'. Up to 16,384 universe IDs can be assigned with a 14-bit field. The size of DMX data ranges from 1 to 512 bytes. As for vulnerable packets, the check sum is used.

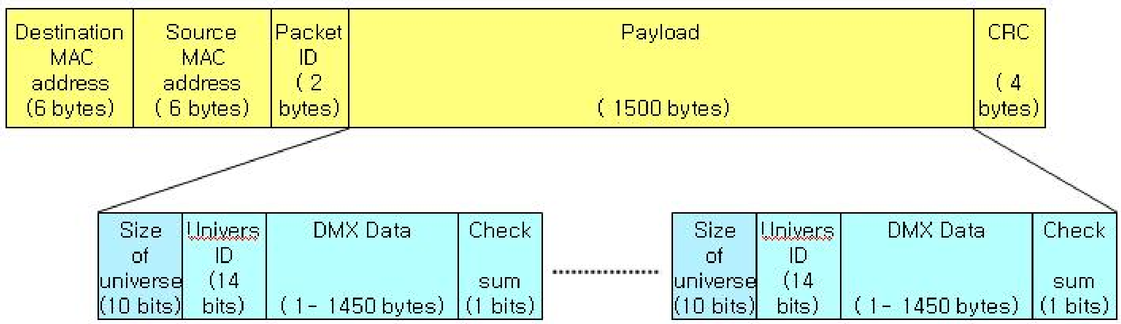


**Figure 3. Encapsulation of a WDMX Packet with Universes**

For diversified lighting effects, lots of DMX channels are helpful and should be transmitted to the DMX devices through a WLAN NIC at the adequate refresh rate by a WDMX server. Even though some lighting performance needs to play few thousands of DMX channels, it is necessary to transmit the whole acting script with less than the Ethernet frame length. As shown in Figure 4, the whole acting script is fragmented into the sequence of WDMX packet not to exceed Ethernet frame of 1,500 bytes. In general, few universes can be included within a packet of 1,500 bytes. The minimum number of universes within a packet becomes 2.907 (1,500/516) and the maximum becomes 300 (1,500/5).

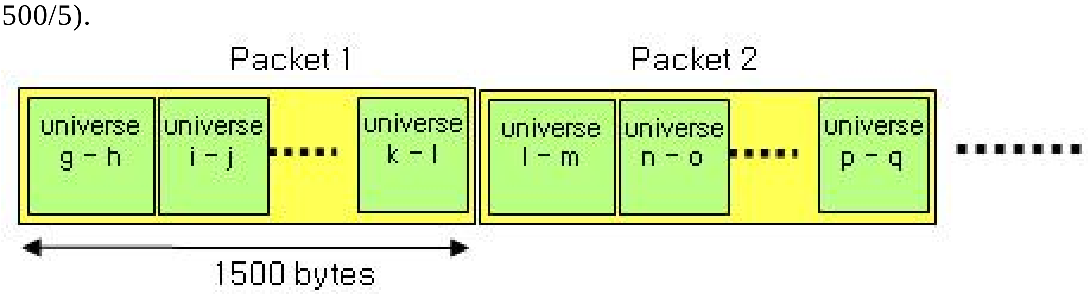


**Figure 4. Performance Acting Script Data Fragmented into 1500 Bytes**

In the actual implementation, the full length of all acting script is assumed to be equal to 10,500 bytes. The packet ID ranges between 0x11 and 0x17. The server transmits repeatedly 7 WDMX packets (10,500 bytes) at the refresh rate of 40 on the broadcast mode and a recipient of WDMX controllers receives pertinent packets.

Receiving WDMX controller can interpret both universe ID value and DMX value. Then, the throughput ratio becomes 3,360,000 bps (10,500 x 40 x 8).

### 3.2. NDIS Miniport Driver

Now, the throughput for the WDMX server should be 3.36 Mbps at least. In general, a server application program to support with 802.11 WLAN is written based on TCP/IP socket or Winsock[12]. Alternatively, a raw socket allows direct access only to the TCP/IP header of a network frame. However, any raw socket or Winsock program does not allow direct access to the Ethernet frame.

Fortunately, Microsoft has developed and released Network Driver Interface Specification (NDIS) technology [13]. NDIS can retrieve data from a physical device for direct input and output. It is possible via NDIS miniport to access a lot of functions of the WLAN card. Our approach is to integrate NDIS with WDMX server software.

The advantage of our technique is that we can control the entire network packet on the WDMX server. To use the entire network packet taking place on the server system, rolling Network Driver Interface Specification (NDIS) protocol driver [13] is the only solution on Windows, especially in Windows XP SP2, Microsoft Vista and Windows 7. The NDIS is a Windows specification as it is a kernel-mode network driver that defines the routines that network drivers should implement [14][15]. The NDIS is a library of functions that forms the upper sub layer of the OSI data link layer [16] and acts as an interface between level 3 network protocol drivers and the hardware level MAC drivers [17].

As shown in Figure 5, the NDIS miniport driver sits in between MAC and IP layer and can control all traffic being accepted NIC card. NDIS 5.x miniport driver that supports wireless LAN frames appears to the operating system. In order to write the server application based on NDIS, Microsoft DDK (driver development kit) is utilized. This NDIS driver converts a WDMX packet into an Ethernet frame. That is to say, the 802.3 MAC header is replaced with the appropriate 802.11 MAC header on transmission. On receipt of packet, NDIS driver replaces the 802.11 MAC header with the appropriate 802.3 MAC header.

In this paper, NDIS 802.11 wireless LAN miniport driver is implemented in order to provide a network interface over the WLAN media for both infrastructure and ad hoc networks. This driver is built on the same NDIS model used for connectionless miniport drivers. After delivering the WDMX packet to the NIC, the miniport driver completes the send operation by calling NDIS in send complete. The server program is written based on MFC.

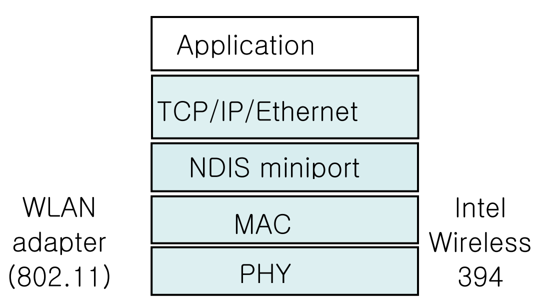


**Figure 5. NDIS WLAN Miniport Driver for the Conversion into Ethernet**

## 4. Experiments and Results

The WDMX server and the WDMX controllers are communicated through wireless LAN in ad-hoc mode. The server transmits WDMX packets in broadcasting mode, and each WDMX controller makes a judgment on receipt of packets. DMX512 frame is retrieved from the received packet before transmission to DMX512 lighting devices. The server application program transmits a lighting direction message to each WDMX controller through a linked wireless LAN card.

For an experimental setup, a commercial LED device, Droplet™ manufactured by Xilver B.V is connected via WDMX controller developed. The LED lighting device is tested in general mode RGB with 12 DMX512 channels.

The server application program written with Visual C++ can send repeatedly WDMX packet data to the WDMX controllers. The total length of a WDMX packet becomes 1514 bytes including MAC address fields. The program can send out 7 WDMX packets of 10,500 (1500 x 7) bytes every 5 millisecond and the elapse time to send out 7 packets is measured. Figure 6 shows experiment results for elapse times of the two server systems. One system is CompaQ Presario 2800 based on Intel Pentium 4 with AcroLan (802.11b). Another server system is CompaQ Presario v3000 based Genuine Intel T2130 with Intel pro/wireless 3945(802.11g).

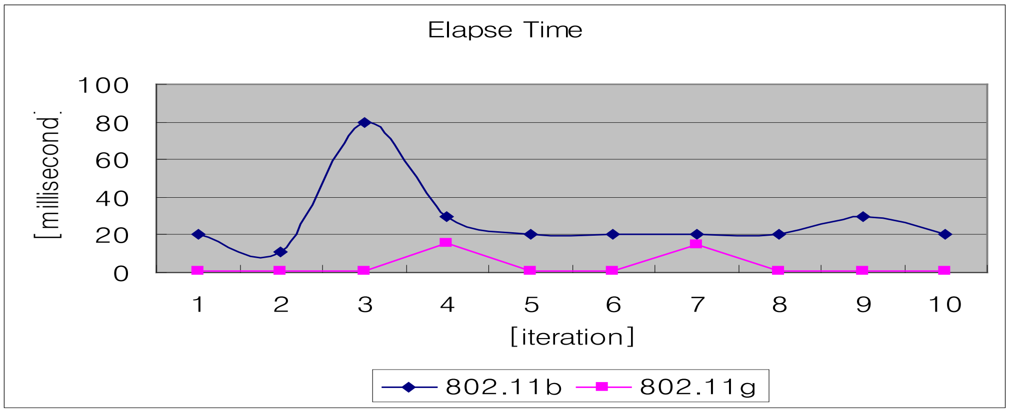


**Figure 6. Elapse Time of WDMX Packet Transmission**

The average of elapse times is measured with 27.1 ms for 802.11b and 3.9 ms for 802.11g, respectively. Then, based on the measured elapse time, an average throughput is calculated with 3,128,561 (1514 x 7 x 8 / 27.1ms) bps and 21,739,487 (1514 x 7 x 8 / 3.9ms) bps, respectively. A thread is issued to secure transmission of WDMX packets every 5 ms. Experiment results of 3,128,561 and 21,739,487 bps are compared with 11 Mbps and 54 Mbps of 802.11 b/g, respectively.

In order to verify the operation of WLAN NDIS miniport, a sending packet from WDMX server is analyzed. Figure 7 shows an Ethernet frame captured and analyzed by Windump program [18]. A WDMX packet is not same with the standard Ethernet frame. The Ethernet type is unknown because WDMX packet is not a standard. From the application of WLAN miniport driver, the WLAN module is treated much as Ethernet module. The length of WDMX packet is 1514 bytes. The packet ID is 11 (11 from 0x5511). The size of a universe cell is 16 bytes and the universe number is 1166 for a DMX device.

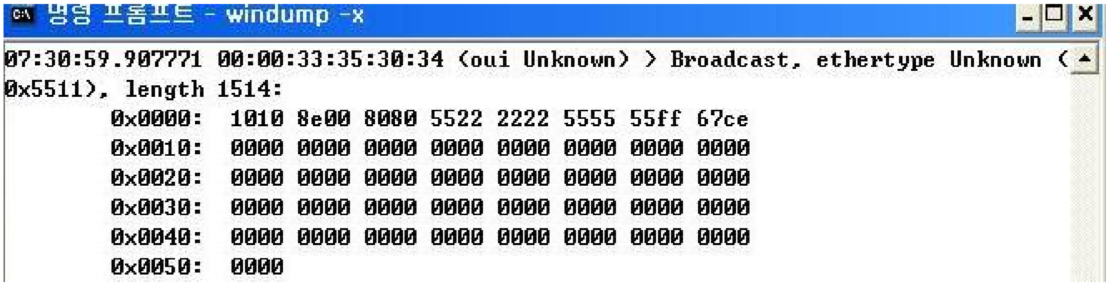


**Figure 7. Packet Verification by Windump**

## 5. Conclusions

This paper has proposed a new WDMX server technology aimed to improve LED lighting effects. In order to increase and expand the number of DMX device through WLAN, WDMX server has been proposed. A new WDMX packet format is designed and implemented. The proposed WDMX packet format can support up to 16,384 ($2^{14}$) universes to generate diversified lighting effects and it supports the compatibility with standard DMX. To secure fast transmission of WDMX packets, WLAN NDIS miniport driver has been proposed and developed. WLAN NDIS miniport can allow direct access to Ethernet frame. The WDMX server does convert WDMX packet into Ethernet frame. WLAN NDIS miniport driver does convert the Ethernet frame into WLAN frame.

The WDMX controller converts the WDMX packet into the DMX frame based on RS-485. After receiving WDMX packet, the WDMX controller transfers that to DMX devices. A lighting device is selected based on universe ID address and DMX address. WDMX packet is captured and analyzed through Windump program. Moreover, the transmission speed of WDMX packet by using NDIS WLAN miniport is estimated to about 3.1 Mbps and 21.7Mbps for 802.11b/g, respectively. Experiment also shows that the WDMX packet received from the server is compatible with the standard DMX protocol.

Last but not least, the WDMX system shows possibilities of new application with more increased DMX devices. The effects of card section can be generated through control of each lighting device in performance locations or other places where the public gathers together.

## Acknowledgements

This work was supported by the GRRC program of Gyeonggi province. [(2012-B5), Situational technology and System development].

## References

[1] http://www.taittechnologies.com.
[2] Web site http://www.esta.org/about/index.html.
[3] MIDI Control of Stage Lighting, First Draft 12/9/05, www.innovateshowcontrols.com.
[4] "Specification for the Art-Net mx Ethernet Communication Standard", Artistic License (UK) Ltd, web site http://www.artisticlicence.com/user-guides/art -net.pdf, **(2004)** June.
[5] S. Shon, J. J. Woo and Y. Han, "Implementation of a Novel WDMX Controller for LED Lights", Journal of the Korean Institute of Illuminating and Electrical Installation Engineers, vol. 22, no. 10, **(2008)**, pp. 1-7.
[6] http://www.goddarddesign.com/wspread.html.
[7] J. I. Kim and B. H. Hwang, "Implementation of Broadband LED Lighting System Using ZDMX Modules Based DMX512 Protocol", Journal of Korea Contents Association (JKCA), vol. 10, no. 11, **(2010)**, pp. 38-47.

[8] http://www.rdmprotocol.org.
[9] http://www.plasa.org.
[10] "USITT DMX512-A - Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories", Entertainment Services and Technology Association, **(2000)**.
[11] http://www.usitt.org/USITT DMX512-A.
[12] G. R. Wright and W. R. Stevens, "TCP/IP Illustrated", The Implementation. Addison-Wesley Professional, vol. 2, **(1995)**.
[13] S. Dhawan, "Networking Device Drivers", John Wiley & Sons Inc., **(1995)**.
[14] W. Oney, "Programming the Microsoft Windows Driver Model", Microsoft Press, **(2003)**.
[15] Microsoft MSDN Library, NDIS Intermediate Driver, Retrieved 20, from Microsoft MSDN Library Web site: http://msdn.microsoft. com/enus/library/ ff557012(v=vs.85).aspx, **(2011)** May.
[16] W. R. Stevens, "TCP/IP Illustrated", The Protocol. Addison-Wesley Professional Computing Series, vol. 1, **(1994)**.
[17] W. R. Stevens, "TCP/IP Illustrated", TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols, Addison-Wesley Professional, vol. 3, **(1996)**.
[18] http://www.winpcap.org/windump.

## Authors

**Sugoog Shon** (Corresponding author ) received his B.S. (1982) degree in Electrical Engineering from Seoul National University, his M.S. (1984) degree in Electrical Engineering from Seoul National University, and his Ph.D. (1996) degree in Electrical and Computer Engineering from the University of Texas at Austin. He is a professor in the Department of Information and Telecommunication at the University of Suwon in Korea. His research interests include computer and embedded system, robot, network protocol, network simulation, and network programming.

**Yifeng Cui** received the B.S. (2003) and M.S. (2007) degree in the College of Computer Science and Technology of Shandong University of Technology, he is a lecturer in the Linyi University of China. Now, he is studying for a doctor's degree in the University of Suwon in Korea. His research interests include computer measurement and control, network engineering and robot control system.

**Heejung Byun** received her B.S degree from Soongsil University, Korea, in 1999, the M.S. degree from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2001, and the Ph.D. degree from KAIST in 2005. She was a senior researcher in Samsung Electronics, Ltd. from 2007 to 2010. She is currently a professor with the Department of Information and Telecommunications Engineering, Suwon University, Korea. Her research interests include network protocol, network modeling, controller design, and performance analysis.